Working Paper no. **110**

Consistent Estimation of Agent Based Models

Jakob Grazzini
Department of Economics "Cognetti de Martiis"
University of Turin

June, 2011

Laboratorio R. Revelli, *Collegio Carlo Alberto*
Via Real Collegio, 30 - 10024 Moncalieri (TO)
Tel. +39 011 670.50.60 - Fax +39 011 670.50.61
www.laboratoriorevelli.it - labor@laboratoriorevelli.it

Consitent Estimation of Agent Based Models

Jakob Grazzini *

Department of Economics "Cognetti de Martiis"
University of Turin

June 16, 2011

Abstract

The aim of this paper is to describe a method to introduce empirical data in agent based models. Starting from the econometric and calibration literature, it is shown how to select the values of the parameters in the model and which conditions has to be met to have consistent estimations. A crucial point lays in the analysis of the artificial data produced by model, in particular to test for ergodicity and stationarity.

Keywords: Agent-based models, Estimation, Calibration, Simulation

1 Introduction

The economic system is composed by many different autonomous agents that interact with each other and with the environment. The result is a system that exhibits emergent properties: the properties at the macro level cannot be explained directly by the properties at the micro level [22]. Agent based modeling is a tool used to overcome the limitations of pure mathematical analysis, it allows the construction of more realistic models; unfortunately this happens at a cost. Indeed, agent based models are more difficult to understand, to generalize and to explain. A model consisting of algebraically solved equations can easily be interpreted and generalized using formal proofs. Despite the fact that it can be considered as a well defined set of equations [34], an agent based model suffers from the different (smaller) degree of knowledge about the functions that are at the base of the model. While analytical results are conditional only in relation to the specific hypothesis about the model, simulation results are conditional in relation to both the specific hypothesis of the model and to the specific values of the parameters used in the simulation runs: each run of an agent based model yields a sufficiency

*I would like to thank Matteo Richiardi. For any comment please write to jakob.grazzini@unito.it

theorem, but a single run does not provide any information on the robustness of such theorems [7]. One way to treat the "sufficiency problem" in agent computing is through multiple runs, systematically varying initial conditions or parameters in order to assess the robustness of results [7]. Another (not alternative) option is to find the most realistic setting of the model and analyze the model in its neighborhood. This paper describes a method to use empirical data in computational models; by using observed data about the system under analysis it is possible to select the values for the parameters so that the artificial data and the observed data are as similar as possible. The "sufficiency theorems" proven by the runs produced with the "data-driven" agent based models are still just sufficiency theorems, but are quite interesting sufficiency theorems since they show the behavior of the model in the neighborhood of the most realistic setting. Introducing empirical data in agent based models is also a fundamental step in the interpretation of the model and possibly in the validation of it [42, 8], but it is still largely missing in the literature [1, 38, 34]. Quantitative methods to make inference on a complex model are very interesting both in finance and in economics. The interaction between the agents in a stock market, for example, is largely accepted as fundamental in shaping the properties of the markets. This led to build (e.g. [5, 31, 35, 10, 13]) and rarely to estimate complex financial models [9, 1, 23], for a survey see [11]. The starting point to find the "data-driven" parameters is the simulation based econometrics literature. The smaller degree of knowledge about the model is a problem also for the estimation procedure. The properties of the model are not know *a priori*, this means that to know how to interpret the parameters resulting from simulation based econometrics methods, the model has to be tested. In particular to know whether the estimation produces consistent parameters, the optimization heuristic has to be tested (to know whether there exists a global minimum and whether the heuristic is able to find it) and there is the need to test for stationarity and ergodicity of the artificial data.

The aim of this paper is to contribute to the empirical research in agent based models by showing how to introduce empirical data into an agent based model, and in particular stressing the importance of the analysis of the artificial data. "Data-Driven" is a general notion to denote the procedure that leads to the selection of parameter values using observational data. The actual interpretation of the data-driven parameters (e.g. as estimated parameters) depends on the properties of the agent based model, in particular on stationarity and ergodicity. In the following sections the "experimental model" used to evaluate the estimation method will be presented, followed by the description of how to estimate the model. Even if the estimation (or "data-driven") is presented in the particular environment defined by the model, the set of proposed tests can be used in any situation. Every model that is stationary, ergodic and that have a global minimum for the defined objective function can be estimated.

2 The Model: El Farol Bar Problem

The model chosen to be the experimental model is the El Farol Bar Problem introduced by [4][1]. This model has been chosen due to its similarity with many economic situations in which the agents make a decision by looking at past events, as in a stock market where part of the information the agents use to decide on their actions is gathered from the past behavior of the price. The model is built in the following way. N agents have to decide whether to go or not to go to the El Farol Bar in Santa Fe (New Mexico). Since the space in the bar is limited, the agents will prefer to go to the bar if the total number of agents attending the bar is below a given threshold; otherwise they will prefer to stay at home. In [4] the total number of agents is 100 and the dimension of the bar is such that the agents consider it too crowded when there are more than 60 agents present. There is no sure way to tell in advance the number of attending people, therefore the agents have to formulate an expectation about the agents that will attend the bar and make a decision upon it. If an agent expects that more than 60 agents will go, (s)he will stay home, otherwise (s)he will decide to go. To formulate the expectations the agents use "bounded rationality" strategies that use past attendance to forecast next period attendance. Every agent has a set of available strategies, and chooses to use the strategy that has performed best in the past periods (for further details see [4]). In this paper the model has been simplified by eliminating the learning mechanism: the agents will not be able to choose a strategy among a set of available strategies but will have just one strategy each. Supposing that the described model is a good representation of a real system, the kind of parameters that will be "data-driven" depends on the assumption and knowledge available about the system. Supposing for example that the structure of the strategies used by the agents and the distribution of the strategies between the agents are known, the interesting parameters would be the ones that define the strategies. For example it may be known that half of the agents decide whether to go or not to go to the bar looking back on the attendance of x_1 weeks ago, and the other half computes a moving average over the last x_2 weeks. In such a case the estimation would be performed for the parameters inside the strategies (x_1 and x_2). In a second case it can be supposed that the strategies are completely known but the distribution of the strategies among the agents is unknown. The parameters to be estimated would thus be the distribution of the strategies among the agents. A third and more complete estimation is the union of the two cases, which means finding both the strategy parameters and the distributional parameters. In this work the parameters that will be found are the parameters that define the distribution of the strategies in the model.

The available strategies at the system level are the following ones:

1. Cyclical: the agent forecasts next week's attendance by looking at the attendance 7 weeks ago. If the forecast is above the threshold, he decides not to go, otherwise he decides to goo

[1] The model reproduced in Python can be found on the internet: http://jakob.altervista.org/PythonElFarol.zip

2. Moving average: the agent computes the average attendance of the last 5 weeks. If the forecast is above the threshold, he decides not to go, otherwise he decides to go

3. Fixed: the agents forecast the same attendance (65 agents), regardless of the past history (they never go!)

The scope is to find a "data-driven" method and to evaluate it. To perform this task a set of "pseudo-real data" will be produced. Since different distributions of the strategies produce different outcomes, if a set of assumptions is satisfied, it will be possible to find the distributional parameters that produce the pseudo-real data. The aim is to replicate a real setting in which a theoretical model and a set of observed data are available. Given the model and given the data, the aim is to find the values of the parameters that minimize the difference between the real data and the artificial data. The starting point is the simulation based econometrics literature and the calibration literature.

3 Estimation, Calibration and "Data-Driven"

In agent based models, and in computational models in general, there are non-linear relations between parameters and emergent outcome of the model. To start an investigation of methods and potentiality of a data driven approach it is possible to refer to the literature on simulation based econometrics and calibration.

Simulation Based Econometrics

Agent based modeling is an instrument used to model complex phenomena that involve interactions between the elements of the system under analysis, interaction between the elements and the environments, heterogeneity and so on. Agent based models are thus used in situations in which the analytical approach is too restrictive to have a good representation of the system. From the previous, obvious statement it follows that it is not possible to use standard econometric tools to compare artificial data and real data. Indeed, the complexity of an agent-based model impedes the writing of an analytical condition to find the parameters that minimize a given distance between real data and the model. To overcome this difficulty it is possible to refer to the literature on simulation-based econometric methods. A good reference in this framework is Gouriéroux and Monfort's book [12] where the most important methods are described: Methods of Simulated Moments (MSM), Indirect Inference and Simulated Maximum Likelihood. In the following, the method of simulated moments will be used; it is an intuitive way to extend simulated econometrics to agent based models. The simulated methods of moments was introduced by McFadden [18] and Pakes and Pollard [36]. Duffie and Singleton [17] apply the method of simulated moments to a markovian process, Lee and Ingram [33] apply it to time series models. Suppose that we have a set of observations y_t , a vector of D explanatory variables $\{\mathbf{z_t}\}$ and a vector of K instruments $\{\mathbf{x_t}\}$. Supposing that a well specified theoretical model is available and that it tells that there is a linear functional relationship between the observations and the explanatory variables $y_t = \mathbf{z_t}'\delta + \epsilon_t$, the aim is to estimate the vector of D parameters. Supposing moreover that the data generator process is well behaved, i.e. that $(y_t, \mathbf{z_t}, \mathbf{x_t})$ are jointly ergodic and jointly stationary and that the orthogonality conditions are satisfied, it is possible to estimate the parameters by minimizing the function $J(\delta, W)$, where $J(.)$ is the quadratic form that represents the distance between the theoretical moments and the real moments (that is between the orthogonality conditions and the sample counterparts) and where δ is the vector of parameters and W is a weighting matrix (see [29]). The hypothes is that the model is well specified implies that the real data are a realization of the theoretical model with an unknown set of parameters (that is what we are looking for) and with a given random sequence. Given the stationarity and ergodicity properties it is possible to estimate consistently the parameters by minimizing the distance between the moments of the real data and the moments of the artificial data conditional to the value of the vector of parameters (that is the conditional moments of the theoretical model). The method of moments or the general method of moments requires the possibility of computing analytically the theoretical moments; unfortunately such condition significantly limits the its applicability. If the model is complex (and non linear), it may be impossible to find an analytical form of the conditional moments and thus it may be impossible to find an analytical expression of the quadratic form and of its derivatives. This means that it may be impossible to minimize analytically the objective function. The solution is to simulate the model: if the theoretical expected moments conditional to the parameters cannot be found, it is possible to simulate the model and compute the moments from the artificial data. The method of simulated moments thus extends the method of moments by replacing the population moments with its simulated counterpart calculated with simulated data [17]. To estimate the parameters it is sufficient to choose the value of the parameters that minimize the distance between the simulated moments and the observed moments. The general expression of the objective function to be minimized can be found in Gourieroux and Monfort ([12], p. 27), where also the asymptotic properties of the estimator are shown. In particular when the number of observations (n) tends to infinity and the number of simulations (S) is fixed, the estimator is strongly consistent and its distribution tends towards a Normal under regularity conditions in [27]. The variance of the simulated moments estimator (given the weighting matrix W) decreases when S increases, and tends to be equal to the variance of the GMM estimator when $S \to \infty$. Indeed, it is shown that for $S \to \infty$ the method of simulated moments estimation is equivalent to the GMM estimation [12]. The extension to agent based models is straightforward: the artificial data produced by the simulation model are used to compute the simulated moments to be compared with the observed moments. The variance of the simulation based estimator depends on the choice of the weighting matrix, on the variance of the random component inside the model and on the randomness due to the simulation procedure. To compute the variance of the estimator and to be able to make hypothesis testing on the resulting parameters it is possible to run the estimation procedure several times and estimate the variance of the estimator using the resulting sample of estimations. The computational model that will be used in this work is particularly simple compared to other agent based models. The agents simply decide their moves based on a strategy that uses the previous periods. There is neither explanatory variables nor instruments. Despite the simplicity, there is no way of writing any analytical expression of the emergent data (the attendance at the bar). Since the model is dynamic and it has no explanatory variable, the simulated moments conditional to are computed by running the model for n periods (where n is the number of observations) computing the first M non-centered moments (provided that they exist), simulate the model S times and compute the average moment over the S simulations (see equation 2). An alternative estimator for the theoretical moments is to compute the simulated moments using one run with Sn observations. Under stationarity and ergodicity the two estomator are both consistent, and under strict stationarity or $n \to \infty$ or $S \to \infty$ they have also the same variance[2]. The advantage of using the first estimator (S simulation of length n) is that it is possbile to compute the moments also in a non ergodic situation. If $S \to \infty$, the simulated moments tend to the theoretical moments and the MSM estimator tends to the GMM estimator. The simulated moments are conditional on the values of the parameters used to run the simulations and the estimation requires the minimization of the distance between the actual moments and the simulated moments. The objective function is:

$$J(\delta, W) = (\mu^R - \mu^S(\delta))' W (\mu^R - \mu^S(\delta)) \tag{1}$$

where μ^R is the vector of dimension M containing the first M non-centered moments computed over the real data, μ^S is the vector of dimension M containing the first M non-centered moments computed over the simulated data, the simulated data depends on the parameters used to run the simulation. The element m of the vector μ^S is:

$$\mu_m^S = \frac{1}{S} \sum_{s=1}^{S} \left(\frac{1}{n} \sum_{t=1}^{n} y_t^m \right)_s \tag{2}$$

The estimated set of parameters is the solution of the minimization of $J(\delta, W)$. As known from [27], the number of moments M in the objective function has to be equal or greater then the number of parameters to be estimated, that is the dimension D of δ. If $M = D$ there is perfect identification and the solution of the minimization is the set of parameters δ_0 such that $J(\delta_0, W) = 0$. Note that this condition holds necessarily only in the case in which $J(\delta, W)$ is continuous in δ, which is not the case in a computational model. Since the simulation can handle only discrete values of δ, it is not always possible to find such that $J(\delta_0, W) = 0$. Under the regularity condition defined in [27], the values of the parameters resulting from the minimization of (1), $\hat{\delta}$, are consistent and normally distribuited [12]. A crucial issue for agent based estimation is to know whether the regularity condition are actually met. In particular this paper will focus on the consistency of the estimator and thus on the stationarity and ergodicity properties.

[2] See Appendix A

In this implementation the model is well specified for construction; in "real" application the model will be supposed to be well specified. Stationarity and ergodicity have to be tested. Nonparametric tests proposed in [26] will be made on the artificial data. This is a crucial difference compared to the standard econometric literature where the stationarity test is made on observed data and ergodicity is supposed. Given the hypothesis of a well-specified model, it is possible to use the model itself to test the properties of the data generator process. This method allows increasing the number of observations to increase the power of the nonparametric test and allows testing for ergodicity, which is usually supposed.

Calibration

"Calibration consists of choosing numerical values for the parameters so that for the existing policy regime, the model reproduces (either exactly or closely) the benchmark data as a model solution" [14], the parameters can be exogenous, chosen from the literature, and/or endogenous, chosen to match the real data. The aim of calibration is to parametrize a model, and the most logical way of choosing parameters is to use the information derived from real data [14]. Given the theoretical models, the task of the calibration is to give answers to the problems of the day, to understand the effects of a given policy, using the numerical implementation [14]. All that is needed to use the models is to choose the appropriate value of the parameters. A very famous and early attempt to calibrate a model can be found in [32], and will then be developed especially in the business cycle literature. In [32] the calibration procedure is used to test the theory; i.e. to control whether there is a set of parameters for which the model is quantitatively consistent with the real data. The calibration consists in choosing most of the parameters in the literature and leaving the other parameters free to be determined using real data. In particular, to choose the value of the parameters, a set of properties (e.g. moments) of the actual data are compared to a set of properties of the artificial data. The differences and similarities between calibration and estimation are well summarized in [37] and in [28] among others. The main distinction lies in the fact that "calibrated models" may use parameters taken from the literature to address the over-parametrization of a model [19]. Since the parameters in the literature have often been found in different economic and theoretical contexts, calibration has to be used carefully. In this paper the distinction between calibration and estimation is not crucial for the results, for this reason the term estimation will be used without distinction.

Data-driven

The previously defined concepts need two fundamental and very rare properties: stationarity and ergodicity. The possibility of building more realistic models using agent based models is directly linked to the very realistic feature of non-ergodicity and/or non-stationarity of the models. Since the problem is very common, it will be central in the following work; while non-stationarity can be reduced to stationarity with traditional methods, non-ergodicity is a major problem. Note that it is very difficult to foresee the behavior of an agent-based model without running it. This means that the agent-based model has to be built using a theoretical and intuitive foundation and cannot be built to be ergodic. In standard economic modeling, stationarity is tested on data while ergodicity is supposed. This latter hypothesis cannot be made a priori in agent-based modeling, the ergodicity of the model must be tested and supposing the good specification of the model, the result of the test can be extended to the real system under analysis. Since many models (real systems) will turn out to be non-ergodic it is useful to define the "data-driven" parameters. Data-driven in this paper is defined in a very broad sense as a method that allows using the observed data when estimation is not possible. Data-driven parameters are simply a set of values of the parameters that minimize the objective function, although they cannot be interpreted as estimations. Estimation is rigorous inference while the data-driven method allows the use of data when estimation would be meaningless. The difference will be only in the assumptions and consequently in the interpretation of the parameters, the use of the real data and the method will be the same. The aim of defining the data-driven parameters is to avoid confusion and explicitly consider the non-ergodicity problem.

To summarize, in order to make inference and to estimate the structural parameters of the model it is necessary to have a well-specified model and an ergodic and stationary data generator process. The model is supposed to be well specified, or at least represent a good approximation of the reality. The ergodicity and stationarity will be tested using nonparametric tests. What is interesting here is to use real data in agent based models, and in particular understand how to interpret the results. If stationarity and ergodicity are not satisfied, the inference about the real system is by definition impossible, but real data can still be used to specify the parameters. Using exactly the same method (same objective function and same optimization heuristic) it is possible to find the value of the parameters that minimize the distance between a given set of moments computed over the real data and the same set of moments computed over the artificial data. The procedure will be called "data-driven"; the interpretation of the resulting parameters as estimation or simply empirically plausible values depends on the properties of the model. In both cases the outcome of the procedure improves the specification of the agent based model and decreases the "sufficiency problem".

4 Objective Function

The aim is to find the set of parameter values that minimize the distance between the real data and the artificial data. The definition of distance is crucial, as different objective functions produce outcomes with different properties. The reference to the simulation based econometric literature is important in order to understand the conditions needed for the consistent estimation of the parameters and the properties of the estimators. The chosen objective function refers to the method of simulated moments (e.g. [12, 17, 33]) and is compatible also with the use of the moments made in the calibration literature [32]: the distance between the model and the real data is measured as the quadratic difference between a given set of moments computed over the real data and the set of moments computed over the artificial data. A similar objective function has been proposed in [43]. The selection of the moments should be such that the objective function is able to discriminate between alternative models and/or parameters; moreover [43] advises the use of a large number of moments in order to increase the probability of identifying the model parameters. Since the aim of the "data-driven" is to mimic the properties of the observed data, the moment based objective function is perfectly compatible. The weighting matrix is important only for the efficiency of the estimations and will not be taken into consideration in this paper (see [43] for more details about the weighting matrix). The choice is to use an identity matrix and focus on the consistency property. Given the assumption that the model is well specified, the moments of the artificial data are a function of both the parameter values and the random seed used to initialize the model. If the random process is stationary and ergodic and if the orthogonality condition is satisfied, it is possible to find consistent estimates of the parameters. The consistency property assures that by increasing the number of observations the estimations tend towards the true value of the parameters. In a traditional econometric setting, the parameters would be found by comparing the moments computed over the real data and the theoretical moments computed using the model. In this framework, this last operation in not possible, and for this reason it is necessary to compute the conditional moments using simulations. The moments computed over a simulation are different from the theoretical moments of the model because a simulation uses a random draw of the random component embedded in the model. By increasing the number of simulations the simulated moments tend towards the theoretical moments, and the MSM estimator tends towards the GMM estimator [12]. The objective function is the equation (1) with the identity matrix as weighting matrix and using the first 10 non-centered moments:

$$J(\delta, W) = \sum_{m=1}^{M} (\mu_m^R - \mu_m^S)^2 \tag{3}$$

where M is equal to 10 and m represents the degree of the non-centered moments,

$$\mu_m^R = \frac{1}{n} \sum_{t=1}^{n} y_t^m \tag{4}$$

y_t is the observation t, n is the number of observations. The simulated moment of order m is the moment of order m computed over a simulation of n observations and it depends on the value of the parameters and on the random seed:

$$\hat{\mu}_m^s(\delta, \epsilon_s) = \frac{1}{n} \sum_{t=1}^{n} \hat{y}_t^m(\delta, \epsilon_s) \tag{5}$$

$\hat{y}_t$ is the simulated observation i, δ is the vector of parameters and ϵ_s is the random sequence used for the simulation. In this particular case the simulation depends only on δ and ϵ_s. To decrease the influence of the random seed on the value of the simulated moments it is possible to compute the moment over a set of S simulations, as noted above for S that tends towards infinity the simulated moments tends towards the theoretical moments:

$$\mu_m^S = \frac{1}{S}\sum_{s=1}^{S}\hat{\mu}_m^s(\delta, \epsilon_s) \tag{6}$$

For example, the simulated mean of the emergent properties of the ElFarol model will be computed by running the model with a given set of parameters and different random seeds 5 times and computing the mean for each simulation and then the mean of the five computed means. Note that the random seeds used for each combination of the parameters has to be the same, otherwise it would be impossible to disentangle the effect on the value of the artificial moments (and in the objective function) due to the change of the random component and to the change of the values of the parameters used in the simulation.

5 Optimization heuristic and genetic algorithm

Given the objective function, the problem is to minimize it. The objective function cannot be written analytically, is not globally convex and has many local minima. The complexity of the model and of the search space requires the use of heuristics to find the minimum of the objective function. The most naïve approach is the brute force approach: by creating a set of all the combinations of the parameters it is possible to run the model for each of these combinations and choose the combination that gives the minimum value of the objective function. Theoretically possible, the brute force approach would be practically impossible in most applications. The use of an optimization heuristic reduces the computing time by searching the space with a given set of rules. The set of rules (i.e. the chosen heuristic) and the objective function are crucial for an efficient optimization. In particular the objective function has to be able to well characterize the results: the more the value of the objective function change by changing the value of the parameter the more the search will be efficient. Increasing the number of moments in the objective function increases the search efficiency and the ability of the optimization heuristic to find the minimum. The optimization function is an instrument, it is a way to find the minimum in complex optimization problems, but it has no influence over the properties of the minimum itself. A heuristic has to be able to provide a good approximation of the global optimum, it cannot be problem-specific and it has to be easily implemented [24]. The advantage of using an optimization heuristic is that it does not need strong hypotheses about the optimization problem, apart from the assumption that a global minimum actually exists. On the other side heuristics cannot produce high-quality solutions with certainty [24], so they have to be used only when a traditional analytical method cannot be applied.

[23] use a combination of the Nelder-Mead simplex direct search method and the threshold accepting optimization heuristic. In this paper a genetic algorithm will be used. Genetic algorithms have been introduced by [30] and have had quite a success in

the economics literature; it has been used for example in [5] as a classifier system and by [6, 3, 2] and [40] as a learning mechanism. The use of a genetic algorithm is motivated by the fact that it is well known, easily implemented and its optimization mechanism is simple. A genetic algorithm mimics the Darwinian evolutionary mechanism. A set of binary strings representing a set of strategies are chosen randomly. The best strings, defined according to a given fitness or an objective function, are selected probabilistically creating a new set of strings with the same cardinality as the original set; the genetic operators will then act on the selected strings, cross-over and mutation. The selection exploits the best strings in the population while cross-over and mutation create diversity and explore the search space. The result is a new set of strings that have to be evaluated: the model is run and the fitness is computed for each string and the cycle starts again. For further details about genetic algorithms see [30], [25], [21]. In this paper the strings represent the values of the three parameters defining the distribution of the strategies among the agents. The fitness of each string is computed using the objective function; the lower the value of the objective function the higher is the probability of being selected. To build a genetic algorithm, the first issue to face is how to encode the parameters of interest into a binary string.

5.1 Encoding the strategies

The first problem is to understand how to encode the parameters: we need a binary string that produces deterministically three natural numbers, and the sum of the three numbers has to be equal to the number of agents in the model. The problem is very similar to the problem faced during the distribution of the seats in parliament after an election; see for example [16] and [15]. In this case the problem is less complex since it is not necessary to bargain with political representatives. The aim is to use a deterministic algorithm that is able to transform a binary string, used by the genetic algorithm, to a string containing the number of agents that uses each strategy. Every genetic string has 7 bits for each of three parameters; the total length of the genetic string is of 21 bits. For example: $(1,1,1,0,0,1,1,0,1,1,1,1,1,1,1,0,0,1,1,0,1)$

To evaluate the performance of a string, it is necessary to translate the binary string into natural numbers representing the distribution of the strategies among the agents. The first step is to compute the decimal numbers corresponding to each of the seven digit binary numbers. The result is a string of three numbers between 0 and 127 (that corresponds to a seven digit binary number with all ones). Transforming the above string, the following vector is obtained:$(115, 63, 77)$

Since the aim is to find the distribution of the strategies in the system, the next operation is to transform the natural numbers in weights. Each number has to be between zero and one and the sum has to be one. The simplest method to obtain a vector of weight from the vector of natural numbers is to divide each element of the above vector by the sum of the three elements in the vector. The position in the vector represents the strategy, then the weight of strategy i, w_i, is:

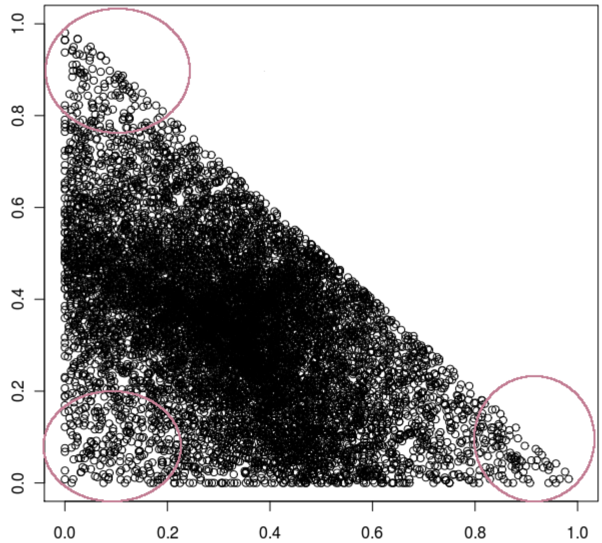

Figure 1: The points represent the resulting weights. The extreme values (selected parts) are less likely.

$$w_i = \frac{x_i}{\sum x_i} \tag{7}$$

where x_i is the element i in the vector. In the vector above the sum of the elements is $\sum xi = 255$ and the resulting string is: $(0.450980392, 0.247058824, 0.301960784)$.

The sum of the elements of this last vector is 1. It is the string of weights of each of the three strategies in the model. Note that this way of computing the weights may be biased [39]. Using this method, the central numbers (for example the weights 0.3, 0.3, 0.4) are more likely because more combinations of the original integer number (between 0 and 127) produce such central numbers. To show the problem 10,000 triples of numbers between 0 and 127 has been randomly created. Following the above procedure, the weight has been computed and the first two elements of the weight vector have been plotted in figure 1: it is evident that there is a cloud in the central part of the plot that is more dense than on the extreme positions. This means that if the real distribution is extreme, it is harder (not impossible) for the genetic algorithm to find it. One solution is to include some deterministic strings in the initial set of random strings that reproduce the extreme positions. In this way it is possible to control the performance of these strings and force them into the population.

To use the weights in the model, the weight has to correspond to the actual number of agents that use the strategy i. The first step is to multiply each weight by the total number of agents in the model (in this case 100) and round it to nearest natural number: $(45.0980392, 24.7058824, 30.1960784)$. Rounding to: $(45, 25, 30)$.

There may still be a problem: the sum of the new natural numbers can be below or above the total number of agents. To solve the problem, a very simple algorithm has been used: if the sum is above 100, the smallest number on the list is decreased by one, if the sum is below 100, greatest element on the list is increased (iterating the algorithm until the sum is 100). Every possible binary string can be transformed deterministically (a given string always gives the same distribution of strategies) in a string containing the number of agents that uses each strategy. This last vector of numbers is used to run the model and obtain the attendance for that distribution of strategies; by computing the moments and by using the objective function the fitness of every given string is obtained. The aim is to find the string (the distribution of strategies) that minimizes the objective function. This naive encoding method is simple to understand and to implement but it is not efficient. If the model is complex and computationally heavy and/or the number of parameters to be found is high, the following problems can slow down the search for the minimum: a) it may happen that strings with a small difference in bits represents a quite different distribution of strategies in the population; b) the space of search is wider than needed, indeed different strings can represent the same distribution of strategies.

Depending on the nature of the variables and on the dimension of the search space it may be worth it to develop alternative ways of encoding the parameters for the genetic algorithm or even choose a different optimization heuristic. In this specific case the main problem derives from the nature of the parameters: since they represent weights they must sum to one.

5.2 Checking the performance of the genetic algorithm

A crucial point is to check whether the chosen heuristic is appropriate for the complexity of the search space. Supposing that a set of real data is available it is important to be sure that, given the complexity of the model and of the search space, the chosen heuristic is able to find the minimum. To make the test, the data-driven method has to be run over a set of pseudo-real data created with a known random seed. Using only one run of the model and the same random seed used to create the data, the minimum value of the objective function is known, equal to zero, and reached with the "pseudo-real parameters" (given the random seed and just one run, the model always produces the same time series with a given set of parameters). This procedure can be thought as a way of understanding the search space. This operation tells how efficient the genetic algorithm is and can help in developing more efficient heuristic and in setting the parameters of the genetic algorithm (as mutation and cross-over probability). The "data-driven" procedure can easily become very heavy since the genetic algorithm will have to run many times many models with different parameters. For this reason it is important to use efficient optimization heuristics. To create a faster genetic algorithm a slightly modified genetic algorithm has been developed.

The problem of a genetic algorithm is that it faces a trade off between exploiting the minimum in the population (i.e. to perform a local search around the minimum

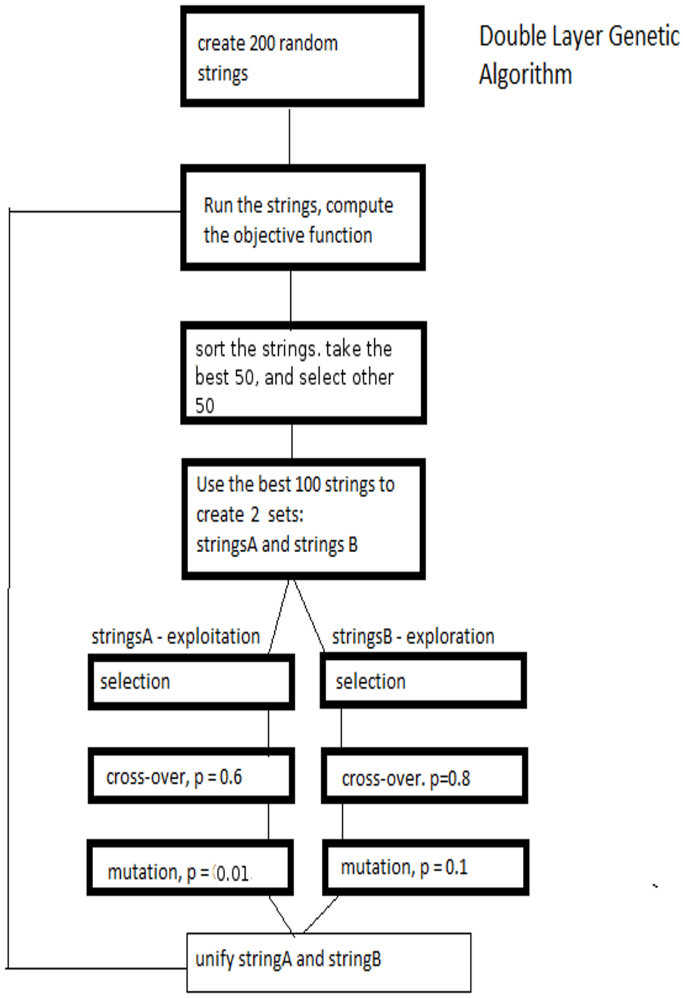

Figure 2: The modified Genetic Algorithm

it has already found achieved with very low cross over and mutation probability) and exploration (which is achieved with high cross over and mutation probability). The aim of the modified genetic algorithm is to decrease this trade off. As an initial set 200 random strings have been chosen. The modified genetic algorithm evaluates the strings, sorts them, takes the best 50 strings and selects probabilistically 50 strings to create a set of 100 strings. It creates two different sets both containing the 100 selected strings and starts the genetic operations on the two sets, one with low mutation and cross-over (respectively 0.01 and 0.6) that exploits the best strings in the set, the second that uses very high mutation and cross-over (0.1 and 0.8). When the genetic operations on the two sets are over, they are unified to recreate a 200 string set and starts over again by selecting the best 100 strings. The idea behind this modified genetic algorithm is to exploit the best strings in the population and at the same time explore the space. If the exploring set finds a good string it will be selected (with probability 1 if it is top 50), and it will be used as a starting string both in the exploiting set and in the exploring set. The modified genetic algorithm can find the minimum in 4.7 (average over 10 trials) genetic generations, while the normal genetic algorithm, run with the same parameters of the exploiting part of the modified algorithm and with 200 strings, takes 8.9 (average over 10 trials with the same random seed of the trials with the modified algorithm) genetic generations. In this simple case the modified genetic algorithm permits saving half of the computing time; the difference is remarkable. When the real procedure will be run, each string will be evaluated using the moments computed over 5 runs of the model. This means that a large amount of computing time will be saved using the modified genetic algorithm.

6 The Genetic Data-Driven Algorithm

It is now possible to understand how the "data-driven" method works. The following steps are crucial:

1. Build the model
2. Build the genetic algorithm and encode the parameters (create the heuristic)
3. Check the search space and the performance of the heuristic
4. Run the calibrator with the real data
5. Test for stationarity and ergodicity in the neighborhood of the "data-driven" parameters

The model is built using the SLAPP protocol[3]. The genetic algorithm has been added above the observer module. The calibrator works in the following way. It starts by creating 200 random strings and distributing them to the observers. The observers translate (deterministically) the strings into the value of the parameters (the weights of

[3]see http://eco83.econ.unito.it/terna/slapp/

the three different strategies) and run the model. The run of the model produces an outcome, the attendance. The observers run the model 5 times, collect the attendance data and compute the simulated moments. It must be stressed that the random seed used by the observers is always the same and is fixed at the start[4]; the reason is that it has to be possible to distinguish between the change of the outcome due to a change of the random seed and a change due to the values of the parameters (the latter is the most interesting). Running the estimation many times with different random seeds gives the possibility of finding the variance of the estimates and eventually making some hypothesis testing. Given the simulated moments the value of the objective function associated with each string is computed to evaluate the performance of each string. The selection will select a set of strings, cross-over and mutation will act on the selected string, and the cycle will start again. The new strings are distributed to the observers, run in the model and fitness is computed. It has been tested above that with a known minimum the genetic calibrator is able to find the minimum. In the following part the method is applied to a more realistic setting where the true model is available but the random seed that produced the pseudo-real data is unknown.

6.1 Data-Driven

To create the pseudo-real data, a random distribution of the strategies has been chosen. There are 100 agents, who forecast next week's attendance by using a simple strategy. If the forecast is above 60 (threshold) agents they do not want to go (they dislike the crowded bar), if the forecast is below the threshold they will decide to go. The strategies are 1) cyclical; 2) moving average; 3) fixed.

The distribution of strategies is: 29 agents use strategy 1, 39 use strategy 2, 32 use strategy 3. The pseudo-real data set is created by letting the El Farol model run for 500 periods as shown in figure 2.

6.2 The parameters

The next step is to use the data-driven method to search for the parameters that produced the pseudo-real data. Following the literature on simulated moment estimation, by increasing the number of observations the consistency of the estimated parameters increases, and by increasing the number of simulation runs the difference between the simulated moment estimator and the GMM estimator is reduced (i.e. the difference between simulated and theoretical moments is reduced). The number of observations is 500, and the number of simulations is 5. The evaluation of a string is done by running the model 5 times. The 5 random seeds used for the 5 runs remain constant. The average value for each moment is computed and compared with the pseudo-real moments through the objective function. As mentioned above this procedure gives consistent estimates only if the data are stationary and ergodic. The data generator process is the

[4]The random seed is fixed when the observer starts to run the models. The result is as if 5 different seeds were fixed at the beginning of each run. These (virtually) 5 different seeds has to be always the same.

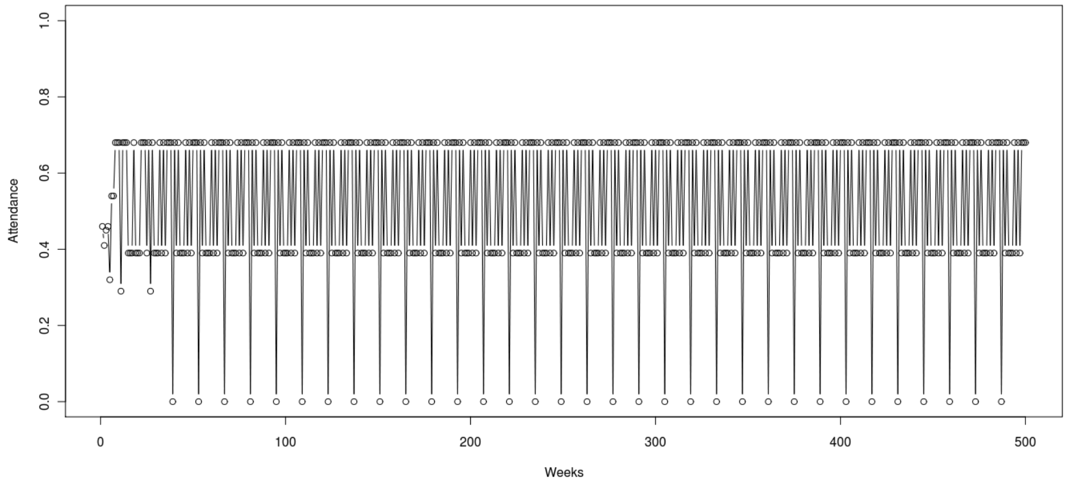

Figure 3: The pseudo real data. This data has been produced with a known distribution of the strategies in the model. The aim will be to find this distribution using the pseudo real data and the model.

model; the actual data that the process generates are random, since they depend on the random seed. Let's define "process moments" as the moments that are computed on a single realization of the process, that is on a time series produced with a given sequence of random numbers (with a given random seed) and "ensemble moments" as the moments computed over more than one realization of the process. A process is stationary when the "process moments" do not depend on the time (week) in which they are computed. A process is ergodic when the process moments tend towards the ensemble moments by increasing the observations. This means that if a process is ergodic and if enough observations of a single process are available, the information about the data generator process is enough to infer its properties. If a process is non-ergodic, one single realization is not sufficient to infer about the data generator process. These two properties are fundamental for every econometric analysis. In the real world they are normally not satisfied, and in particular ergodicity is a very rare property. In standard econometrics, stationarity is tested while ergodicity is supposed. The reason is that it is not possible to test ergodicity on real data, for the obvious reason that only one observation is available. In agent based models it is usually not possible to predict the properties of the emergent data of a model, for this reason once the model has been built it is necessary to test for stationarity and ergodicity. This will be done following the method outlined in [26]. The outcome of the tests tells whether the parameters found by minimizing the objective function can be interpreted as estimation or simply as calibrated values.

6.3 Stationarity and Ergodicity

To check the stationarity and ergodicity of the data generator process the tests proposed in [26] will be used. The behavior of an agent based model is a priori unknown and needs to be tested. The aim is to estimate some structural parameters by comparing the moments of a real time series and the artificial moments computed using the agent based models. Since the model represents a complex system, or at least a system in which the emerging behavior is not reducible to an analytical form, the researcher has no knowledge about the distribution of the random term or the effects of the randomness on the emerging properties; for these reasons a non-parametric tests must be used. Supposing that the model is well specified, the usual lack of power of non-parametric tests is overcome by testing not the real time series but the artificial time series. The idea is that if the model represents the reality, or at least an approximation of it, it is possible to suppose that the model has the same properties of the real system under analysis. This assumption permits the use of a non-parametric test over an elevated number of observations produced by the model, instead of using real observations that are usually costly and few. An agent based model can exhibit very different behaviors by simply changing the value of the parameters. This is also the case with the (modified) El Farol Bar Problem. In particular it will be seen that with different values of the parameters (which in this case represent the distribution of the strategies among the agents) the model can produce an ergodic or non-ergodic behavior. In a real setting the strategy should be to run the "data-driven" procedure and then test the ergodicity and stationarity in the neighborhood of the estimated parameters. In the present framework, to make the analysis simpler, the stationarity and ergodicity are tested with the pseudo-real parameters. The stationarity test described in [26] checks whether the moments are constant during a given time series using the Runs Test (or Wald-Wolfowitz test) proposed in [41]. Since all the agents choose deterministically, given the past behavior of the attendance, once all the agents have enough observations to use their strategy, the series becomes deterministic. The only random part lies in the starting condition, as the agents use a random behavior only when they are not able to use their strategy. The behavior of the attendance is cyclical, and the moments will not change over time. Given that the series is deterministic it is not possible to use the two tailed Run Test, since it would reject the hypothesis of randomness. The constancy of the moments is tested using the one-tailed Run Test (see [20]), for which the alternative is the presence of a trend. The scope of the test is to understand whether the moments are able to characterize the data generator process or not, so in this particular case the interest lies in the constancy rather than in the randomness. The stationarity test is made using a series of 1000 observations divided in 100 windows of length 10. The number of windows is essential for the power of the test, while the length of the windows influences the ability of the test of detecting asymptotic stationarity. To save computational time it is possible to start with many small windows; if the result is stationarity the process is stationary. If the outcome of the test is non-stationarity the test has to be run with longer windows and a longer series (to keep the number of the windows constant) to check whether the process is really non-stationary or it is "asymptotically stationary" .

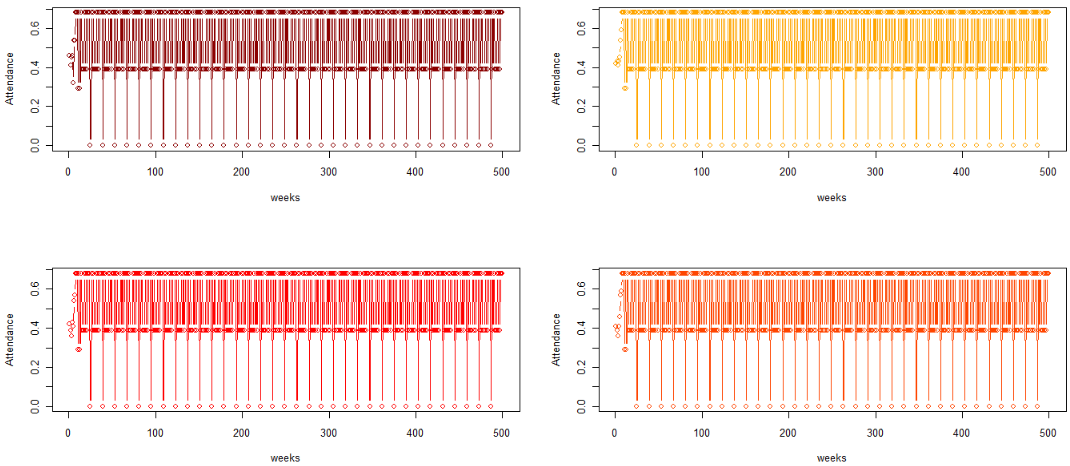

Figure 4: Four different run of the model with different random seed. The model has the same behavior in each run.

The test made on the attendance time series cannot reject the null-hypothesis of constancy of the 10 moments. Each of the 10 moments used in the objective function is stationary, in the sense that they are constant during the time series. The test shows that the moments are not randomly distributed around the overall mean, as the number of runs is too high. The augmented Dickey-Fuller confirms the stationarity result by rejecting the hypothesis of unit root. To be able to make inference from the observed data and to estimate consistently the parameters, also the ergodicity property is needed. A stochastic system is called ergodic if it tends in probability towards a limiting form that is independent of the initial conditions. To be clear, with stationarity the moments are constant *whithin* the series, with ergodicity the moments are constant *between* the series. To test the ergodicity of the data produced by the model it is necessary to test whether the moments produced by different processes tend towards the same value. The agent based model can be considered a function [38, 34], and in this very simple case it is a deterministic function. It is possible to check intuitively that the model is ergodic simply by checking the attendance in a given period. For example in the particular case under exam for 100 different processes the attendance in week 100 is always the same and equal to 0.39 (39 agents at the bar). Since the process is deterministic (apart from a small random initial part) the moments computed on different run of the model are the same. In figure 4 the model has been run with four different random seed, in figure 5 the 10 moments used for the data-driven procedure computed over 30 different run of the model are shown.

To confirm ergodicity it is possible to use the test described in [26] and use the Wald-Wolfowitz test again. The non-rejection of the stationarity hypothesis implies that it is possible to consider the moments as constant in time. Since the process is stationary, the moments computed over the windows created from one long time series (100 windows long

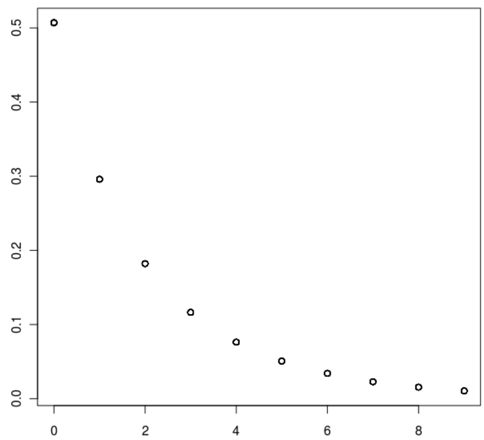

Figure 5: The 10 non centered moments has been computed over each of 30 different run of the model. The value of the moments is always the same.

10 observations in this case) come from the same distribution. If the process is ergodic, the moments computed over 100 processes of the same length of the windows come from the same distributions as the window moments. To test the null of ergodicity the Run Test is used again. A problem arises since the series is deterministic. Indeed, to build the second sample (of processes) the test produces 100 small processes and computes the moment on that processes. If the process is deterministic this procedure will produce a set of almost all equal moments. The fact that the process is deterministic, stationary and ergodic implies that in a given interval of time (e.g. between week 10 and week 20) the attendance will be always the same. The ergodicity test will fail simply because the windows moments are all around the overall mean, while the processes mean will all be the same. To avoid the problem it is possible to slightly modify the ergodicity test to introduce randomness. Instead of always taking the first weeks of the process it is possible to create long processes and use the moments computed over a window of the needed length but taking randomly the starting period. The ergodicity test has been modified in the following way. A time series is produced and 100 windows with given length and random position are chosen from it. The moments computed over these windows compose the first sample in the Wald-Wolfowitz test. The second step is to produce 100 processes (of the same length as the previous one) and choose one window with the given length and random position from each time series, compute the moments and create the second sample of moments. Under the null-hypothesis of ergodicity, the two samples come from the same population, as the random procedure to choose the windows is the same both for the first and second sample. If the process is ergodic then also the set of possible windows that can be chosen is the same both for the first and

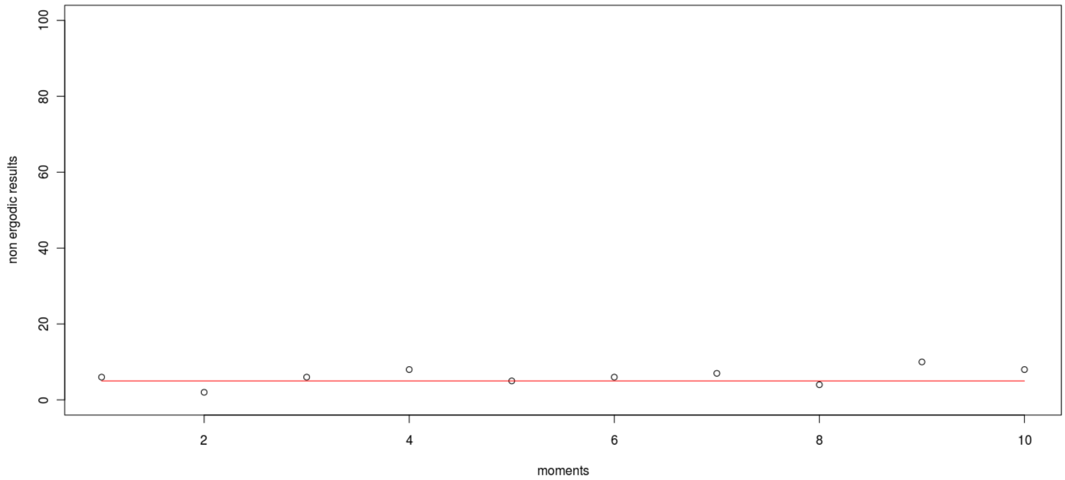

Figure 6: The ergodicity test reults for each moment. The number of non ergodic results over 100 processes. The theoretical number of rejection is 5%. The actual number of rejection for each moment is distributed around 5%

for the second sample. To be clear, if the process is ergodic it is equivalent to choose 100 windows randomly from one time series or to choose 100 windows from 100 different processes. If the process is ergodic, the moments computed randomly on a process come from the same distribution, while if the process is non-ergodic there should be differences between the moments. The test is very similar to the test made in [26], but adds a random component in order to use the test on a deterministic series. The Run Test made 100 times tells us that the two samples really come from the same distribution; the process can be regarded as ergodic since the test gives about 5% of non-ergodicity results (see figure 6), which is the expected value given the chosen type I error ($\alpha = 0.05$).

Given the properties of the model the value of the parameters that will be found using the "data-driven" procedure can actually be considered as estimates of the true parameters. Note that in a real setting the procedure should be inversed, once the data-driven method finds the parameters it is possible to test for stationarity and ergodicity in the neighborhood of the parameters to know how to interpret the data-driven results, that is whether to consider the values as estimations or simply data-driven parameters. In the table 1, the estimated values are listed. In this particular case the values of the estimated parameters are exactly the pseudo-real values, and there is no difference between the estimations with different random seeds. This comes from the fact that the model produces deterministic values and that the series is quite long. For a stochastic process, provided that it is ergodic and stationary, the differences between the estimated parameters with different random seeds can be used as the variance of the estimator and to find the distribution of the estimates to make hypothesis testing. In this particular

seeds	Minimum value of the objective function found by the calibrator	Parameters with minimum objective function
123	3,72E-008	29,39,32
1234	6,40E-008	29,39,32
123456	2,04E-007	29,39,32
654321	2,08E-007	29,39,32
321	9,62E-007	29,39,32
4321	6,30E-008	29,39,32
6543	1,32E-008	29,39,32
78960	1,77E-008	29,39,32

Table 1: The result of the estimation. The column seeds shows the seed used to run the model (has to be constant during the procedure). The second column shows the value of the objective function in the found minimum. The third column shown the value of the parameters that minimizes the objective function.

case the estimated parameters can be considered as a degenerate random distribution with variance zero.

For every seed the calibrator minimizes the objective function and finds the pseudo-real parameters very quickly, in less than 10 generations.

6.4 The non ergodic case

Unfortunately ergodicity is neither a necessary nor a common property; on the contrary it is quite rare in the real world. By taking for example the same identical model with a different distribution of strategies, in particular reducing the weight of the fixed strategy, ergodicity disappears. With a set of parameters equal to 71,22,7 the process exhibits different behaviors depending on the random seed. The stationarity test cannot reject the null-hypothesis of stationarity, while the ergodicity test gives the following non-ergodic results for each of the ten moments over 100 tests: 80, 96, 97, 98, 96, 99, 100, 100, 100, 98. The outcome clearly suggests non-ergodicity. The reason for which there is not full power of the test is that the model produces a finite number of different patterns. In the set of possible attendance behaviors, there is one pattern that is more likely. This means that if the long process used to build the first sample has the most common behavior, and/or by chance many of the processes used to build the second sample are of the most likely type, the test has difficulty in detecting the non-ergodicity. The problem is model specific. In figure 7 the first moment of 100 different run of the model has been plotted. It is clear that there are different regimes at which the model runs, but one of them is more likely to happen.

The number of tests that give non-ergodicity results is enough to infer non-ergodicity. For a further confirmation, figure 8 shows four examples of the model with different random seeds: the behavior of the attendance is clearly different.

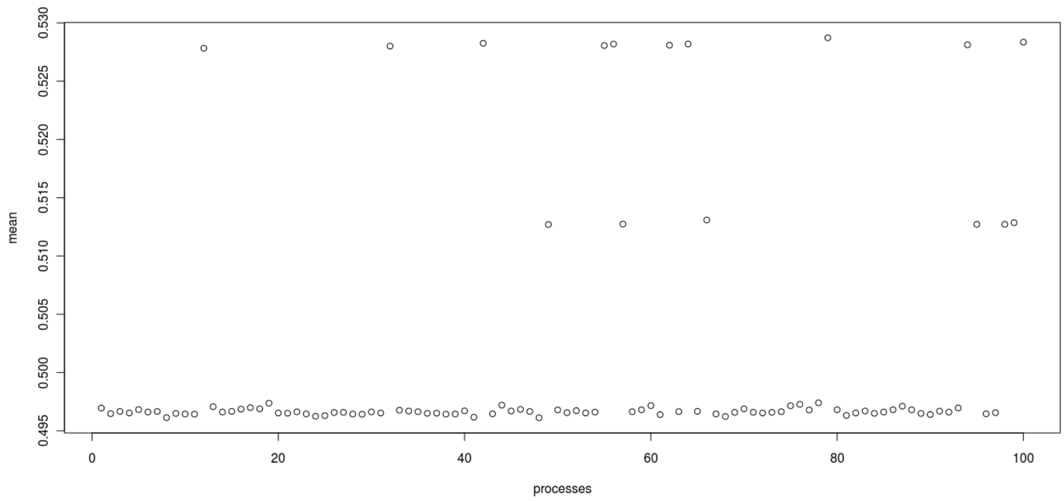

Figure 7: The first moment of 100 different run of the model. The model can run at different regimes, but one of them is more likely

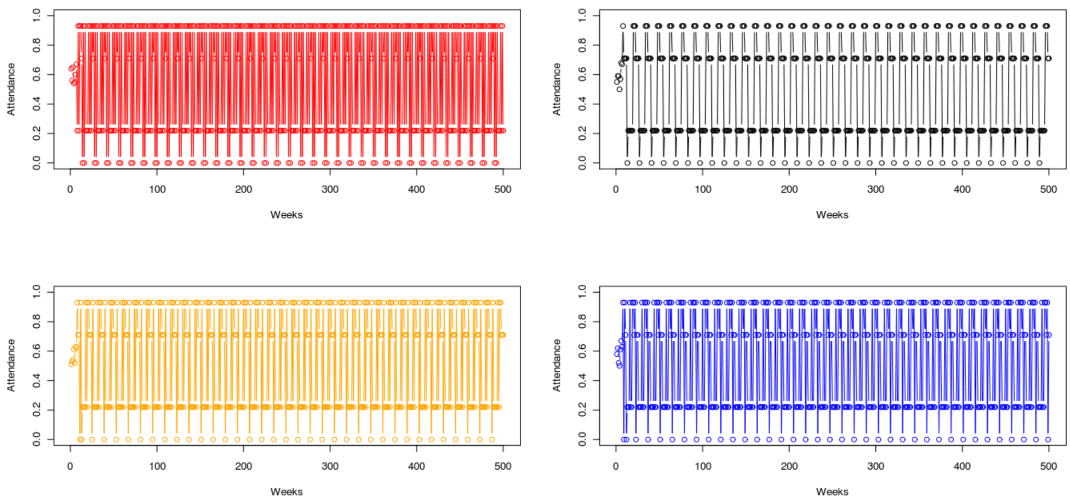

Figure 8: Four different run of the model with four random seeds

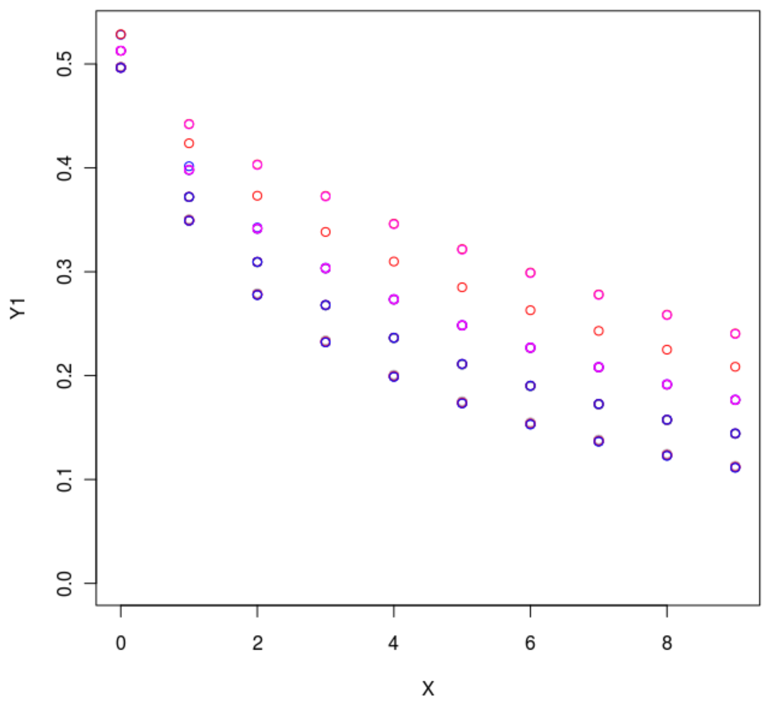

Figure 9: The 10 moments computed over each of 30 run of the model. The same model produces with different seed 3 different means, several second moments etc.

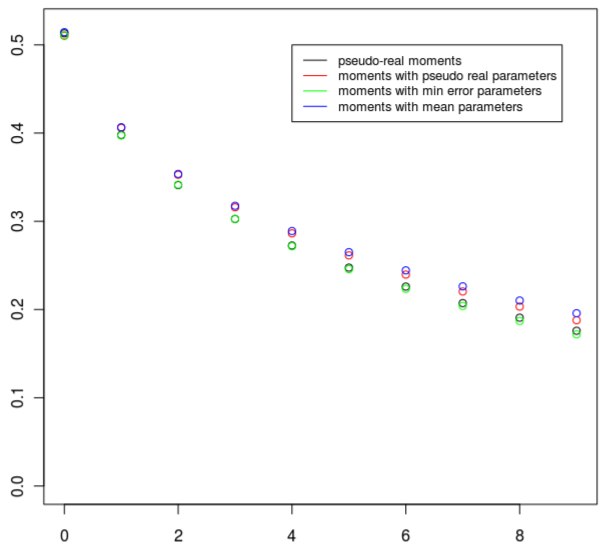

Figure 10: The value of the pseudo real moments and the moment produced by the model with the pseudo real parameters, with the parameters that minimize the objective function, and the mean parameters computed over the results of several data driven procedures.

These different behaviors are reflected in the moments, as it can be seen from figure 9. The moments are different depending on the random seed. This behavior is detected also by the ergodicity test. Non-ergodicity makes inference impossible. Since the random seed is unknown the estimation method will search for the set of parameters that in average produces the minimum value of the objective function. The problem is that the parameters that will be found by minimizing the objective function will not be a consistent estimation of the real parameters; the values will be nothing more than the ones that minimize the objective function. The true parameters produce on average a set of moments that are different from the moments that are produced by a single process (e.g. the one observed). This difference in the moments is systematic. By increasing the number of observations the “process moments” will not tend towards the “ensemble moments”.

Figure 10 shows the problem. The black dots are the moments of the pseudo-real data set. The green dots are the moments (average over five runs) of the best performing set of parameters (74, 21, 5), the red dots are the moments (average over five runs) of the pseudo-real set of parameters. In the presence of non-ergodicity, minimization of the objective function yields a set of parameters that produces on average a set of moments close to the pseudo-real moments, but this set will usually not be the pseudo-real set.

Observer seed	Minimum value of the objective function found by the calibrator	Parameters with minimum objective function
123	0,00019243243863479000	73, 20, 7
1234	0,00038833837065173600	77, 18, 5
123456	0,00019832325097753700	74, 21, 5
654321	0,00020477454266708400	74, 21, 5
321	0,00019464529892941500	69, 25, 6
4321	0,00036185495068086900	67, 27, 6
6543	0,00013306977365778400	75, 18, 7
78960	0,00005624573601091320	74, 19, 7

Table 2: The result of the estimation. The column seeds shows the seed used to run the model (has to be constant during the procedure). The second column shows the value of the objective function in the found minimum. The third column shown the value of the parameters that minimizes the objective function.

The different attendance patterns are enough to bias the estimated parameters. In table 2 the results of the data-driven procedure are shown; the (modified) genetic algorithm ran for 30 periods and has been initialized with 200 strings. The method can find the set of parameters that minimize the objective function but the found values are different from the "pseudo-real parameters"; by increasing the number of observations it would not be possible to achieve an improvement in the estimation of the variables.

Even if the parameters cannot be estimated, it can be useful to compare the model to real data. It is crucial to interpret the values in the right way. The values to be chosen for the parameters could be the mean for each parameter: 72.875, 21.125, 6, that can be rounded to $(73, 21, 6)$ or the values that gives the minimum error, in this case seed 78960 and parameters 74, 21, 5.

If the properties of stationarity and ergodicity are satisfied it is possible to estimate the parameters consistently and efficiently using the method above (given a well specified model), otherwise the best achievement is to reproduce the properties of the observed data set and choose the parameters and the run that give the smallest possible value for the objective function, that is the smallest possible distance between the real moments and the artificial ones. It is crucial to use the resulting parameters not as estimated parameters; the basic idea is that it is better to have no (or little) information instead of wrong information. There may also be the option of reducing noise, and with some luck at the same time get ergodicity. In this particular model the agents use past data for decision making. This is a common characteristic for many economic and particularly financial models. In El Farol, this is a very strong characteristic, as given the past attendance the agents' decisions are deterministic, and all the noise of the model is in the first weeks where the agents do not have enough data to use their strategies. The idea to reduce the noise in such a situation is to use the pseudo-real data to start the model and avoid the initial randomness. The agents instead of deciding randomly, decides using as information the real data. This is coherent with what has been done here: use at best the limited information available. Indeed, the only information is the pseudo-real data and the structure of the model and the aim is to find the parameters of the distribution of the strategies in the population of agents. Given the strategies, it is known that from the first to the ninth week there is noise in the model, while after the ninth week the attendance is deterministic and depends on the past attendance. If the model starts at the tenth week using pseudo-real data as past data the result is a deterministic model perfectly able to reproduce the behavior of the pseudo-real data using the pseudo-real value of the parameters. In this case the real parameters can be found even in a non-ergodic situation. This is a particular case, usually there will probably be noise also after the first weeks, but the idea is that by using observed initial conditions, some noise can be eliminated. Given the data and the model, the use of the initial real data can be used as initial conditions to reduce the noise and obtain better results.

7 Conclusions

This paper presents a method for selecting the parameter values in an agent based model by using observed data. The experimental model is a widely known model in the literature that mimics many economic situations in which the agents use strategies basing decisions on past data. After the model was built it was analyzed in order to be able to use properly the information that can be gathered from the data-driven procedure. The first step is to build an optimization heuristic. Given the heuristic, the search space has to be analyzed in a known situation. This analysis has to be done with pseudo-real data. Performing the minimization of the chosen objective function when the minimum is known, gives important information about the existence of a global minimum (i.e. identification) and about the ability of the heuristic in finding it. Once the heuristic has been proven to work, it is possible to perform the real "data-driven" procedure, which consist in the minimization of a defined distance between real data and artificial data. The last step is to understand how to interpret the value of the parameters resulting from the minimization. Using a stationarity test and an ergodicity test it is possible to determine whether the found values are estimated or simply values that make the model mimic the real system. The difference is crucial and the comparison is useful both in the case in which ergodicity and stationarity are satisfied and in the case in which they are not, but the information and the interpretation of the parameters are radically different. If the model is non ergodic it is possible to reduce the noise using as starting condition the real data. The algorithm can be used also as a tool to select the value of the parameters so that the artificial data has some desired properties. Supposing that the model has been estimated, calibrated or "data-driven" it is possible to use the same method to select policies or parameters, other than the estimated ones, so that the emergent time series has some given properties. For example, in the El Farol Bar problem the dimension of the bar, which is the threshold the agents use to decide whether to go or not to go, can be considerd as a policy instrument. The parameters will be estimated with the current observed policy and given the estimated parameters it is possible to use the data-driven method to select the value of the threshold that minimizes the volatility. The idea is similar to the behavior search proposed by Wilensky in NetLogo[5]. Given the strategies of the agents and their distribution it is possible to optimize the objective function of the policy maker.

Appendix A
Properties of the simulated estimators

Consider a model that given δ produces a time series $\{y_t\}$ (where the dependence on δ is implicit). The aim is to compute the theoretical moments using simulated data. It is possible to use two different estimator:

$$\bar{\mu}_m^S = \frac{1}{S}\sum_{s=1}^{S}\left(\frac{1}{n}\sum_{t=1}^{n} y_t^m\right)_s \tag{8}$$

$$\hat{\mu}_m^S = \frac{1}{Sn}\sum_{t=1}^{Sn} y_t^m \tag{9}$$

where n is the number of observations, S is the total number of simulations and s is simulation s. Supposing that the process $\{y_t\}$ is stationary and ergodic it is possible to show that both estimators are consistent. Indeed, supposing that $E(y_t^m) = \mu^m$, then

$$E\left(\frac{1}{Sn}\sum_{t=1}^{Sn} y_t^m\right) = \mu^m \tag{10}$$

and

$$E\left(\frac{1}{S}\sum_{s=1}^{S}\left(\frac{1}{n}\sum_{t=1}^{n} y_t^m\right)_s\right) = E\left(E\left(\frac{1}{n}\sum_{t=1}^{n} y_t^m\right)\right) = \mu^m \tag{11}$$

This is a direct consequence of the stationarity and ergodicity properties. The ergodic theorem states that the estimators will converge almost surely to their expected value as $n \to \infty$. Note that if the ergodicity property is not satisfied then the two estimators has two different expected values, (8) tends to the overall expected value, while (9) tends to the particular expected value of the particular process (determined by the random initial conditions).This is the reason for which (8) has been used in the paper.

Since the variance of the estimator of the theoretical moments enters directly in the variance of the simulated moments estimator, it is interesting to understand whether one of the two estimators is more efficient. Supposing that the variance of y_t^m is σ_m^2 if the process is *strictly* stationary, then

$$Var\left(\frac{1}{Sn}\sum_{t=1}^{Sn} y_t^m\right) = \frac{1}{(Sn)^2}\sum_{t=1}^{Sn} Var(y_t^m) = \frac{\sigma_m^2}{Sn} \tag{12}$$

[5] see http://www.behaviorsearch.org/index.html

which is the variance of estimator(9). The variance of estimator (1) is

$$Var\left(\frac{1}{S}\sum_{s=1}^{S}\left(\frac{1}{n}\sum_{t=1}^{n}y_t^m\right)_s\right)=\frac{1}{S^2}\sum_{s=1}^{S}Var\left(\frac{1}{n}\sum_{t=1}^{n}y_t^m\right)_s \tag{13}$$

given that the proces is strictly stationary and ergodic we know that for all s

$$Var\left(\frac{1}{n}\sum_{t=1}^{n}y_t^m\right)=\frac{\sigma_m^2}{n} \tag{14}$$

Since each simulation is independent from each other for construction, it is possible to use again the property of the variance operator, that the variance of a sum of independent random variables is the sum of the variances, and by substution (7) in (6) find that

$$\frac{1}{S^2}\sum_{s=1}^{S}\frac{\sigma_m^2}{n}=\frac{\sigma_m^2}{Sn} \tag{15}$$

Since (5) and (15) are equal, we know that under strict stationarity and ergodicity the two estimators are equivalent. If the process is weakly stationary than also the covariances have to be considered. In this case the estimator (9) has the following variance:

$$Var\left(\frac{1}{Sn}\sum_{t=1}^{Sn}y_t^m\right)=\frac{1}{(Sn)^2}\sum_{i}^{Sn}\sum_{j}^{Sn}Cov(y_i^m,y_j^m) \tag{16}$$

while the estimator 8:

$$Var\left(\frac{1}{S}\sum_{s=1}^{S}\frac{1}{n}\sum_{t=1}^{n}y_t^m\right)=\frac{1}{S^2}\sum_{s=1}^{S}Var\left(\frac{1}{n}\sum_{t=1}^{n}y_t^m\right)_s \tag{17}$$

using the fact that each simulation is independent. Supposing that the process is ergodic, thus that for each s the variance of the process is the same than it is possible to substitute

$$Var\left(\frac{1}{n}\sum_{t=1}^{n}y_t^m\right)=\frac{1}{n^2}\sum_{i}^{n}\sum_{j}^{n}Cov(y_i^m,y_j^m) \tag{18}$$

in equation (17), to eventually find the variance of estimator (1)

$$\frac{1}{S^2}\sum_{s=1}^{S}Var\left(\frac{1}{n}\sum_{t=1}^{n}y_t^m\right)_s=\frac{1}{Sn^2}\sum_{i}^{n}\sum_{j}^{n}Cov(y_i^m,y_j^m) \tag{19}$$

The problem is to understand whether (16) and (19) are different. Note that the covariance matrix in (16) is bigger (more arguments in the summation) but it is multiplied by $\frac{1}{(Sn)^2}$ which is smaller than $\frac{1}{Sn^2}$. To know with certainty which of the two estimators is more efficient we should know how the covariance matrix is composed. Let's rewrite

(16) to highlight the part in common with (19). We can divide the summation in (16) in two parts:

$$A = \sum_{s=1}^{S} \sum_{i=1+(s-1)n}^{sn} \sum_{j=1+(s-1)n}^{sn} Cov(y_i^m, y_j^m) \tag{20}$$

$$B = 2\sum_{s=1}^{S-1} \sum_{i=1+(s-1)n}^{sn} \sum_{j=1+sn}^{Sn} Cov(y_i^m, y_j^m) \tag{21}$$

such that (16) is equal to

$$\frac{1}{(Sn)^2}(A + B) \tag{22}$$

Suppose to divide the covariance matrix $Sn \times Sn$ defined in (16) in $n \times n$ sub matrices and call such submatrices N_{ij}. The original covariance matrix, supposing that $S = 3$ can then be represented in the following way:

$$\begin{pmatrix} N_{11} & N_{12} & N_{13} \\ N_{21} & N_{22} & N_{23} \\ N_{31} & N_{32} & N_{33} \end{pmatrix}$$

(20) represents the summation of the elements on the main diagonal N_{11}, N_{22}, N_{33} (the summation of each element the sub-matrices on the main diagonal), while (21) represents the sum of the remaining elements. Since the process is stationary, the covariance between two elements depends only on the distance between the two elements and not on the specific position, which implies that the covariance matrix is symmetric. Since the elements in N_{gh} when $g = h$ are equal, then (20) can be rewritten as S times N_{gg}. Taking for example $s = 1$, (20) can be rewritten as:

$$A = S\sum_{i=1}^{n}\sum_{j=1}^{n} Cov(y_i^m, y_j^m) \tag{23}$$

that is SN_{11}. (23) multiplied by $\frac{1}{(Sn)^2}$ is equal to (19). The difference between (19) and (16) thus is:

$$\frac{1}{(Sn)^2}B \tag{24}$$

The question is whether B is positive, negative or zero, to answer the question it would be necessary to know the elements of the covariance matrix. In general it is not possible to decide which of the two estimators is the most efficient but it is possible to state some important asymptotic properties of the estimators. Recall that the stationarity and ergodicity assumption implies that $\lim_{\tau \to \infty} Cov(X_t, X_{t+\tau}) = 0$. Given the asymptotic behavior of the covariance it is easy to show that when $n \to \infty$ and/or $S \to \infty$ the variance of both estimators tends to zero (which implies also that they tend to be equal).

The fact that the covariance tends to zero as the distance between the observations tends to infinity implies that the summation of the elements of the covariance matrix as the dimension of the matrix increases tend to a finite number. While the summation converge, the denominator in (19) and in (16) instead tends to infinity, that implies that (19) and (16) tend to zero as $n \to \infty$ and/or $S \to \infty$.

To conclude, it has been shown that both estimators are consistent and if the process is strictly stationary, they have also the same variance. If the process is only weakly stationary then the two estimators have different variance. In any case, the variance of the estimators tends to 0 as $S \to \infty$.

References

[1] Simone Alfarano, Thomas Lux, and Friedrich Wagner. Estimation of agent-based models: The case of an asymmetric . *Computational Economics*, 26:19–49, 2005.

[2] Jasmina Arifovic. Genetic algorithms and inflationary economies. *Journal of Monetary Economics*, 36:219–243, 1995.

[3] Jasmina Arifovic, James Bullard, and John Duffy. The transition from stagnation to growth: An adaptive learning approach. *Journal of Economic Growth*, 2(2):185–209, 1997.

[4] W. Brian Arthur. Inductive reasoning and bounded rationality. *American Economic Review*, 84(2):406–411, 1994.

[5] W. Brian Arthur, John H. Holland, Blake LeBaron, Richard Palmer, and Paul Tayler. *The Economy as an Evolving Complex System II*, chapter Asset Pricing under Endogenous Expectations in an Artificial Stock Market. Addison-Wesley, 1997.

[6] Robert Axelrod. *Genetic Algorithms and Simulated Annealing*, chapter The Evolution of Strategies in the Iterated Prisoner's Dilemma. : Pitman e Los Altosy, 1987.

[7] Robert Axtell. Why agents? on the varied motivations for agent computing in the social sciences. *Center on Social and Economic Dynamics Working Paper*, (17), 2000.

[8] Carlo Bianchi, Pasquale Cirillo, Mauro Gallegati, and Pietro A Vagliasindi. Validating and calibrating agent-based models: A case study. *Computational Economics*, 30:245–264, 2007.

[9] H. Peter Boswijk, Cars H. Hommes, and Sebastiano Manzan. Behavioral heterogeneity in stock prices. *Journal of Economic Dynamics and Control*, 31:1938–1970, 2007.

[10] William A. Brock and Cars H. Hommes. Heterogeneous beliefs and routes to chaos in a simple asset pricing model. *Journal of Economic Dynamics and Control*, 22:1235–1274, 1998.

[11] Shu-Heng Chen, Chia-Ling Chang, and Ye-Rong Du. Agent-based economic models and econometrics. *The knowledge Engineering Review*, pages 1–46, 2009.

[12] Alain Monfort Christian Gouriéroux. *Simulation-Based Econometric Methods*. Oxford University Press, New York, 1996.

[13] Rod Cross, Michael Grinfeld, Harbir Lamba, and Tim Seaman. A threshold model of investor psychology. *Physica A*, 354:463–478, 2005.

[14] Christina Dawkins, T.N. Srinivasan, and John Whalley. *Hanbook of Economics 5*, chapter 58. Elsevier, 2001.

[15] Persi Diacons and David Freedman. On rounding percentage. *Journal of the American Statistical Association*, 74(366):359–364, 1979.

[16] Gregor Dorfleltner and Thomas Klein. Rounding with multiplier methods: An efficient algorithm and applications in statistics. *Statistical Papers*, 40:143–157, 1999.

[17] Darrell Duffie and Kenneth J. Singleton. Simulated moments estimation of markov models of asset prices. *Econometrica*, 61(4):929–952, 1993.

[18] Daniel Mc Fadden. A method of simulated moments for estimations of discrete response models without numerical . *Econometrica*, 57(5):995–1026, 1989.

[19] Giorgio Fagiolo, Alessio Moneta, and Paul Windrum. A critical guide to empirical validation of agent-based models in economics: Methodologies, procedures, and open problems. *omputational Economics*, 30:195–226, 2007.

[20] Jean Dickinson Gibbons. *Nonparametric Statistical Inference*. Marcel Dekker Inc., New York, second edition, 1985.

[21] Nigel Gilber and Klaus G. Troitzsch. *Simulation for the social scientist*. University of Michigan Press, second edition, 2005.

[22] Nigel Gilbert. *Modelling and Simulation in the Social Sciences from the Philosophy of Science Point of View*, chapter Holism, Individualism and Emergent Properties. An Approach from the Perspective of Simulation. Elsevier, 2001.

[23] Manfred Gilli and Peter Winker. A global optimization heuristic for estimating agent based models. *Computational Statistics and Data Analysis*, 42:299–312, 2003.

[24] Manfred Gilli and Peter Winker. Review of heuristic optimization methods in econometrics. *Comisef Working Paper Series*, 2008.

[25] David E. Goldberg and John H. Holland. Genetic algorithms and machine learning. *Machine Learning*, 3(2-3), 1988.

[26] Jakob Grazzini. Analysis of the emergent properties: stationarity and ergodicity. *Mimeo*, 2010.

[27] Lars Peter Hansen. Large sample properties of generalized method of moments estimator. *Econometrica*, 50(4):1029–1054, 1982.

[28] Lars Peter Hansen and James J. Heckman. The empirical foundations of calibration. *The Journal of Economic Perspectives*, 10(1):87–104, 1996.

[29] Fumio Hayashi. *Econometrics.* Princeton University Press, Priceton, 2000.

[30] John H. Holland. *Adaptation in Natural and Artificial Systems.* Open University Press, Ann Arbor, 1975.

[31] Alan Kirman. Ants, rationality, and recruitment. *The Quarterly Journal of Economics*, 108(1):137–56, February 1993.

[32] Finn E. Kydland and Edward C. Prescott. Time to build and aggregate fluctuations. *Econometrica*, 50(6):1345–1370, 1982.

[33] Bong-Soo Lee and Beth Fisher Ingram. Simulation estimation of time-series models. *Journal of Econometrics*, 47:197–205, 1991.

[34] Roberto Leombruni and Matteo Richiardi. Why are economists sceptical about agent-based simulations? *Pshysica A*, 3559:103–109, 2005.

[35] Thomas Lux and Michele Marchesi. Scaling and criticality in a stochastic multi-agent model of a financial market. *Nature*, 397:498–500, 1999.

[36] Ariel Pakes and David Pollard. Simulation and the asymptotics of optimization estimators. *Econometrica*, 5:1027–1057, 1989.

[37] Danny T. Quah. Business cycle empirics: Calibration and estimation. *The Economic Journal*, 105(433):1594–1596, 1995.

[38] Matteo Richiardi, Roberto Leombruni, Nicole J. Saam, and Michele Sonnessa. A common protocol for agent-based social simulation. *Journal of Artificial Societies and Social Simulation*, 9(1):15, 2006.

[39] Noah A. Smith and Roy W. Tromble. Sampling uniformly from the unit simplex. *Department of Computer Science, Center for Language and Speech Processing, Johns Hopkins University*, 2004.

[40] Nicolaas Vriend. An illustration of the essential difference between individual and social learning. *Journal of Economic Dynamics and Control*, 24:1–19, 2000.

[41] A. Wald and J. Wolfowitz. On a test whether two samples are from the same population. *The Annals of MAthematical Statistics*, 11(2):147–162, 1940.

[42] Paul Windrum, Giorgio Fagiolo, and Alessio Moneta. Empirical validation of agent-based models: Alternatives and prospects. *Journal of Artificial Societies and Social Simulation*, 10(2):8, 2007.

[43] Peter Winker, Manfred Gilli, and Vahidin Jeleskovic. An objective function for simulation based inference on exchange rate data. *Journal of Economic Interaction and Coordination*, 2:125–145, 2007.